Pascal II: Library units (1 of 2)

"Modular programming" is defined as the separation of procedures and functions, or groups of them, from the main program. Source language modules, called "units", are incorporated into libraries for use with Pascal programs. Units may consist of procedures and functions, or a combination of them written in Pascal or assembly language.

Compiling routines seperately--a necessary part of modular programming--affords you a major advantage in the development of your program: it allows you to approach the task as a group of smaller tasks linked together in a logical manner. The host program must contain a USES statement in order to use routines from the unit.

The two principal kinds of units are called "Regular" and "Intrinsic". NOTE: You are not afforded the capability of having separate units in Apple Pascal.

Regular Units:

When a host program "uses" a regular unit, the Linker physically inserts the unit's code into the host's codefile. Once linked, the files don't need to be relinked until either the unit or the host program is modified and recompiled.

Regular units, since they become part of the host file, may have references to file names. A regular unit may use an intrinsic unit or another regular unit. NOTE: Regular units in version 1.0 of the Apple Pascal system are unable to use intrinsic units.

Install your Regular units in SYSTEM.LIBRARY or in any other library file. Once installed in an alternate library, the Uses statement should include the compiler option $U <library name> before the unit name.

Intrinsic Units:

An intrinsic unit is pre-linked--that is, it contains sufficient information to allow the host program to use it without invoking the Linker. The code for an intrinsic unit remains in the SYSTEM.LIBRARY and is loaded into memory before the host program begins its actual execution. This keeps the size of the host program down; it also allows you to modify and recompile the unit and host program individually without relinking them.

Intrinsic units must be installed in SYSTEM.LIBRARY. Intrinsic units may use other intrinsic units; they may NOT, however, use a regular unit.

NOTE: Intrinsic units may not reference files, such as data files, in Pascal version 1.0.

Assembly Routines as part of Units:

Assembly language routines may be placed into library units. With intrinsic units, the unit is compiled, the machine language routines are assembled, and then the assembled code is linked to the unit prior to installation of the unit into SYSTEM.LIBRARY.

Regular units may also contain machine language routines; however, these routines are not linked to the unit before it's installed in the library. Instead, the host program, the unit and the assembly routines are linked together at the same time.

Additional Notes on the Construction of a Unit:

Any unit which does not contain at least one procedure in the Interface section cannot have an Implementation section. In such a case, however, do include the initialization section--that is, the BEGIN and END.

Procedures listed in the Interface section are "public" to the host program as well as to the unit. Procedures listed only in the Implementation section cannot be accessed by the calling program; such procedures are called "private." When procedures are not listed publicly, they cannot be called from the host; furthermore, the Implementation section is not allowed.

Any intrinsic unit containing a global variable, either public ones (defined in the Interface) or private ones (defined in the Implementation), must have a data segment. If there is no data segment, a system error will occur.

General Format of Units:

The following example is designed to illustrate the general structure of a unit. The line numbers at the left of the page are for reference: they are NOT part of the actual structure.
Apple Tech Notes

Keywords: <None>

==========